**ВВЕДЕНИЕ**

Задание разделено на два этапа:

* Первый этап создание API с использованием PHP фреймворка Laravel и базы данных MySQL
* Второй этап разработка страниц сайта в виде SPA приложения с использованием HTML/CSS и фрейворка Vue, использующего API, разработанного на первом этапе.

**БАЗА ДАННЫХ**

Таблица **users**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| fio | varchar | 150, NOT NULL |
| birthday | date | NULL |
| gender\_id | tinyint | Foreign key, NOT NULL |
| email | varchar | 50, NOT NULL |
| password | varchar | 255, NOT NULL |
| created\_at | datetime | NOT NULL |
| deleted\_at | datetime | NULL |

Таблица **genders**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | tinyint | Primary key, autoincrement |
| name | varchar | 10, NOT NULL |

Таблица **categories**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| name | varchar | 150, NOT NULL |
| parent\_id | int | Foreign key, NULL |
| deleted\_at | datetime | NULL |

Таблица **films**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| name | varchar | 150, NOT NULL |
| country\_id | smallint | Foreign key, NOT NULL |
| duration | smallint | UNSIGNED, NOT NULL |
| year\_of\_issue | year | NOT NULL |
| age | tinyint | UNSIGNED, NOT NULL |
| link\_img | varchar | 255, NULL |
| link\_kinopoisk | varchar | 255, NULL |
| link\_video | varchar | 255, NOT NULL |
| created\_at | datetime | NOT NULL |
| deleted\_at | datetime | NULL |

Таблица **countries**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | smallint | Primary key, autoincrement |
| name | varchar | 50, NOT NULL |

Таблица **categories\_films**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| category\_id | int | Foreign key, NOT NULL |
| film\_id | int | Foreign key, NOT NULL |

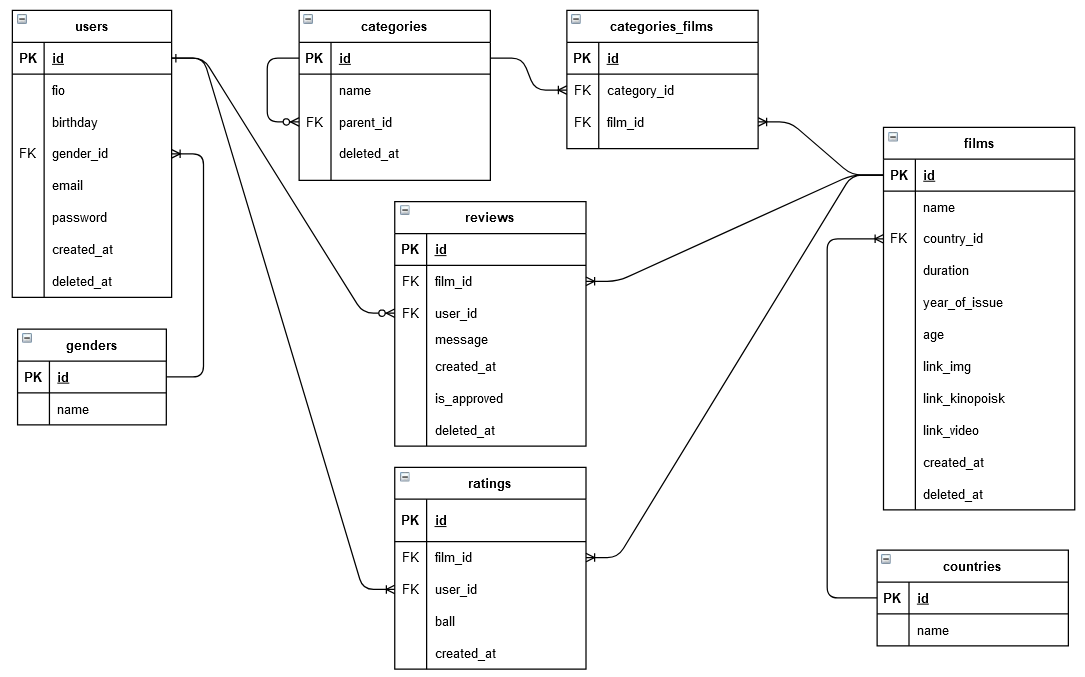
Таблица **reviews**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| film\_id | int | Foreign key, NOT NULL |
| user\_id | int | Foreign key, NOT NULL |
| message | text | NOT NULL |
| created\_at | datetime | NOT NULL |
| is\_approved | tinyint | NOT NULL, DEFAULT 0 |
| deleted\_at | datetime | NULL |

Таблица **ratings**

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип** | **Опции** |
| id | int | Primary key, autoincrement |
| film\_id | int | Foreign key, NOT NULL |
| user\_id | int | Foreign key, NOT NULL |
| ball | tinyint | NOT NULL, UNSIGNED |
| created\_at | datetime | NOT NULL |

**ER-диаграмма**



**1 ЭТАП REST API**

Эндпоинты REST API разделены на открытые (общедоступные) и закрытые, для которых требуется аутентификация.

Общая информация:

* Тела ответов в спецификации содержат некоторые статические примеры данных. Должны использоваться динамические данные из базы данных.
* Параметры заполнителей в URL находятся в фигурных скобках (например {film-id}).
* Порядок свойств в объектах не имеет значения, но порядок в массивах важен.
* Заголовок Content-Type в ответе всегда должен быть application/json, если явно не указано другое значение.
* Временные метки должны соответствовать формату ISO-8601.

Например 2032-01-31T21:59:35.000Z.

* Указанные URL-адреса относятся к базовому URL-адресу API. Например /api/v1/films это URL, чтобы получить все фильмы.
* URL-адреса API не должны заканчиваться .php или .html или любого другого расширения файла.
* Токен для закрытых эндпоинтов должен передаваться как Bearer токен в заголовке `Authorization`. Т.е `Authorization: Bearer <token>`

**Спецификация открытых эндпоинтов**

**GET /api/v1/films**

Возвращает список фильмов с возможностью пагинации, фильтрации и сортировки.

Query Parameters:

| **Параметр** | **Описание** | **По умолчанию** |
| --- | --- | --- |
| page | Номер страницы. Начинается с 1. | 1 |
| size | Количество элементов. Должно быть больше или равно 1. | 10 |
| sortBy | Поле для сортировки. Должен быть один из "name", "year", "rating" | name |
| sortDir | Направление сортировки. Должен быть один из "asc" или "desc" | asc |
| category | Фильтр по жанрам фильмов. Передается id жанра. | 0 |
| country | Фильтр по странам фильмов. Передается id страны. | 0 |

Описание полей сортировки:

| **Поле** | **Описание** |
| --- | --- |
| name | Название фильма |
| year | Год производства фильма |
| rating | Считает среднее арифметическое значение рейтинга по каждому фильму и сортирует по этому значению. |

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"page": 1,

"size": 1,

"total": 15,

"films": [

{

"id": 1,

"name": "Black Mirror",

"duration": 150,

"year\_of\_issue": 2010,

"age": 18,

"link\_img": null,

"link\_kinopoisk": null,

"link\_video": "https://www.imdb.com/title/tt2085059/?ref\_=ext\_shr\_lnk",

"created\_at": "2023-01-31T21:59:35.000Z",

"country": {

"id": 1,

"name": "Pakistan"

},

"categories": [

{

"id": 1,

"name": "Fantasy"

},

{

"id": 5,

"name": "Drama"

}

],

"ratingAvg": 4.5,

"reviewCount": 10

}

]

}

Описание полей в теле ответа:

| **Поле** | **Описание** |
| --- | --- |
| page | Запрашиваемый номер страницы. Начинается с 1. |
| size | Фактический размер страницы. Должен быть меньше или равен, чем запрошенный размер страницы. |
| total | Общее количество элементов независимо от страницы. |
| films | Массив фильмов на странице. |
| ratingAvg | Среднее арифметическое значение рейтинга фильма |
| reviewCount | Количество отзывов к фильму |
| categories | Массив жанров фильма. |

ПРИМЕЧАНИЕ 1: Можно рассчитать, сколько страниц есть:

pageCount = ceil(total / requestedSize)

Он также может быть вычислен, если возвращаемая страница является последней страницей, умножая (page+1) по запрошенному размеру страницы и проверяя, меньше ли результата или равна общим элементам.

isLastPage = (page + 1) \* requestedSize >= total

ПРИМЕЧАНИЕ 2: Если нет изображения или ссылки на кинопоиск, поле равно null.

**GET /api/v1/films/{id}**

Возвращает фильм по id.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"id": 1,

"name": "Black Mirror",

"duration": 150,

"year\_of\_issue": 2010,

"age": 18,

"link\_img": null,

"link\_kinopoisk": null,

"link\_video": "https://www.imdb.com/title/tt2085059/?ref\_=ext\_shr\_lnk",

"created\_at": "2023-01-31T21:59:35.000Z",

"country": {

"id": 1,

"name": "Pakistan"

},

"categories": [

{

"id": 1,

"name": "Fantasy"

},

{

"id": 5,

"name": "Drama"

}

],

"ratingAvg": 4.5,

"reviewCount": 10

}

#### **Ответ при неверном id:**

Status Code: 404

**Response Body:**

{

"message": "Film not found"

}

**GET /api/v1/categories**

Возвращает список доступных жанров.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"categories": [

{

"id": 1,

"name": "Fantasy",

"parentCategory": null,

"filmCount": 10

},

{

"id": 3,

"name": "Western",

"parentCategory": {

"id": 2,

"name": "Action"

},

"filmCount": 15

}

]

}

Описание полей в теле ответа:

| **Поле** | **Описание** |
| --- | --- |
| filmCount | Количество фильмов в жанре |

**GET /api/v1/countries**

Возвращает список доступных стран.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"countries": [

{

"id": 1,

"name": "Pakistan",

"filmCount": 10

}

]

}

Описание полей в теле ответа:

| **Поле** | **Описание** |
| --- | --- |
| filmCount | Количество фильмов выпущенных страной |

**GET /api/v1/genders**

Возвращает список полов.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"genders": [

{

"id": 1,

"name": "Мужской"

},

{

"id": 2,

"name": "Женский"

}

]

}

**GET /api/v1/films/{film-id}/reviews**

Возвращает список одобренных отзывов к фильму.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"reviews": [

{

"id": 1,

"user": {

"id": 1,

"fio": "Ivanov Ivan"

},

"message": "Very Good!",

"created\_at": "2023-02-15T21:59:35.000Z",

}

]

}

#### **Ответ при неверном film-id:**

Status Code: 404

**Response Body:**

{

"message": "Film not found"

}

**POST /api/v1/auth/signup**

Этот эндпоинт создает нового пользователя и возвращает токен сеанса.

**Request Body**:

{

"fio": "Ivanov Ivan",

"email": "ivanov@ivan.kz",

"password": "asdf1234",

"birtday": "2006-04-15",

"gender\_id": 1

}

| **Свойство** | **Ограничения** |
| --- | --- |
| fio | required, min length 2, max length 150 |
| email | required, email, unique, min length 4, max length 50 |
| password | required, min length 6, max length 2^16 |
| birtday | required, date |
| gender\_id | required, exists |

**Response**:

#### **Successful creation response:**

Status Code: 201

Response Body:

{

"status": "success",

"token": "xxx",

"id": 1,

"fio": "Ivanov Ivan"

}

**POST /api/v1/auth/signin**

Этот эндпоинт проверяет email и пароль для всех пользователей. Если обнаружен, возвращается токен сеанса.

***Успешный ответ:***

Status Code: 200

**Request Body**:

{

"email": "ivanov@ivan.kz",

"password": "asdf1234"

}

| **Свойство** | **Ограничения** |
| --- | --- |
| email | required, email, unique, min length 4, max length 50 |
| password | required, min length 6, max length 2^16 |

**Response Body:**

{

"status": "success",

"token": "xxx",

"id": 1,

"fio": "Ivanov Ivan"

}

***Ответ при неверном email/password:***

Status Code: 401

**Response Body:**

{

"status": "invalid",

"message": "Wrong email or password"

}

**Спецификация закрытых эндпоинтов**

**POST /api/v1/auth/signout**

Удаляет токен текущего сеанса.

***Успешный ответ:***

Status Code: 200

**Response Body:**

{

"status": "success"

}

**GET /api/v1/users/{id}**

Возвращает данные пользователя.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"id": 1,

"fio": "Ivanov Ivan",

"email": "ivanov@ivan.kz",

"birthday": "2006-04-15",

"gender": {

"id": 1,

"name": "Мужской"

},

"reviewCount": 85,

"ratingCount": 105,

}

Описание полей в теле ответа:

| **Поле** | **Описание** |
| --- | --- |
| reviewCount | Количество оставленных отзывов |
| ratingCount | Количество оставленных оценок |

**PUT /api/v1/users**

Этот эндпоинт позволяет пользователю обновить свои личные данные.

**Request Body**:

{

"fio": "Ivanov Ivan",

"email": "ivanov@ivan.kz",

"birtday": "2006-04-15",

"gender\_id": 1

}

| **Свойство** | **Ограничения** |
| --- | --- |
| fio | required, min length 2, max length 150 |
| email | required, email, unique, min length 4, max length 50 |
| birtday | required, date |
| gender\_id | required, exists |

**Response**:

#### **Успешный ответ на обновление:**

Status Code: 200

Response Body:

{

"status": "success"

}

**DELETE /api/v1/users**

Пользователь удаляет свой аккаунт.

**Response**:

***Успешный ответ на удаление:***

Status Code: 204

Возвращает пустое тело. Заголовок `Content-Type` не обязательно должен быть `application/json`.

**POST /api/v1/users/{user-id}/reviews**

Этот эндпоинт добавляет новый отзыв пользователя к фильму

**Request Body**:

{

"film\_id": 1,

"message": "Very Good!"

}

| **Свойство** | **Ограничения** |
| --- | --- |
| film\_id | required, exists |
| message | required, min length 4, max length 1024 |

**Response**:

#### **Successful creation response:**

Status Code: 201

Response Body:

{

"id": 1,

"film": {

"id": 1,

"name": "Black Mirror"

},

"message": "Very Good!",

"is\_approved": 0,

"created\_at": "2023-02-15T21:59:35.000Z"

}

**GET /api/v1/users/{user-id}/reviews**

Возвращает список отзывов пользователя к фильмам.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"reviews": [

{

"id": 1,

"film": {

"id": 1,

"name": "Black Mirror"

},

"message": "Very Good!",

"is\_approved": 1,

"created\_at": "2023-02-15T21:59:35.000Z"

}

]

}

#### **Ответ при неверном user-id:**

Status Code: 404

**Response Body:**

{

"message": "User not found"

}

**DELETE /api/v1/users/{user-id}/reviews/{id}**

Пользователь удаляет свой отзыв к фильму.

**Response**:

***Успешный ответ на удаление:***

Status Code: 204

Возвращает пустое тело. Заголовок `Content-Type` не обязательно должен быть `application/json`.

#### **Ответ при неверном user-id:**

Status Code: 404

**Response Body:**

{

"message": "User not found"

}

#### **Ответ при неверном id:**

Status Code: 404

**Response Body:**

{

"message": "Review not found"

}

**POST /api/v1/users/{user-id}/ratings**

Этот эндпоинт добавляет для пользователя новую оценку фильма

**Request Body**:

{

"film\_id": 1,

"ball": 5

}

| **Свойство** | **Ограничения** |
| --- | --- |
| film\_id | required, exists |
| ball | required, min 1, max 5 |

**Response**:

#### **Successful creation response:**

Status Code: 201

Response Body:

{

"id": 1,

"film": {

"id": 1,

"name": "Black Mirror"

},

"score": 5,

"created\_at": "2023-02-15T21:59:35.000Z"

}

***Ответ, если уже есть оценка к фильму:***

Status Code: 401

**Response Body:**

{

"status": "invalid",

"message": "Score exist"

}

**GET /api/v1/users/{user-id}/ratings**

Возвращает список оставленных оценок пользователя к фильмам.

#### **Успешный ответ:**

Status Code: 200

Response Body:

{

"ratings": [

{

"id": 1,

"film": {

"id": 1,

"name": "Black Mirror"

},

"score": 5,

"created\_at": "2023-02-15T21:59:35.000Z"

}

]

}

#### **Ответ при неверном user-id:**

Status Code: 404

**Response Body:**

{

"message": "User not found"

}

**DELETE /api/v1/users/{user-id}/ratings/{id}**

Пользователь удаляет свою оценку к фильму.

**Response**:

***Успешный ответ на удаление:***

Status Code: 204

Возвращает пустое тело. Заголовок `Content-Type` не обязательно должен быть `application/json`.

#### **Ответ при неверном user-id:**

Status Code: 404

**Response Body:**

{

"message": "User not found"

}

#### **Ответ при неверном id:**

Status Code: 404

**Response Body:**

{

"message": "Rating not found"

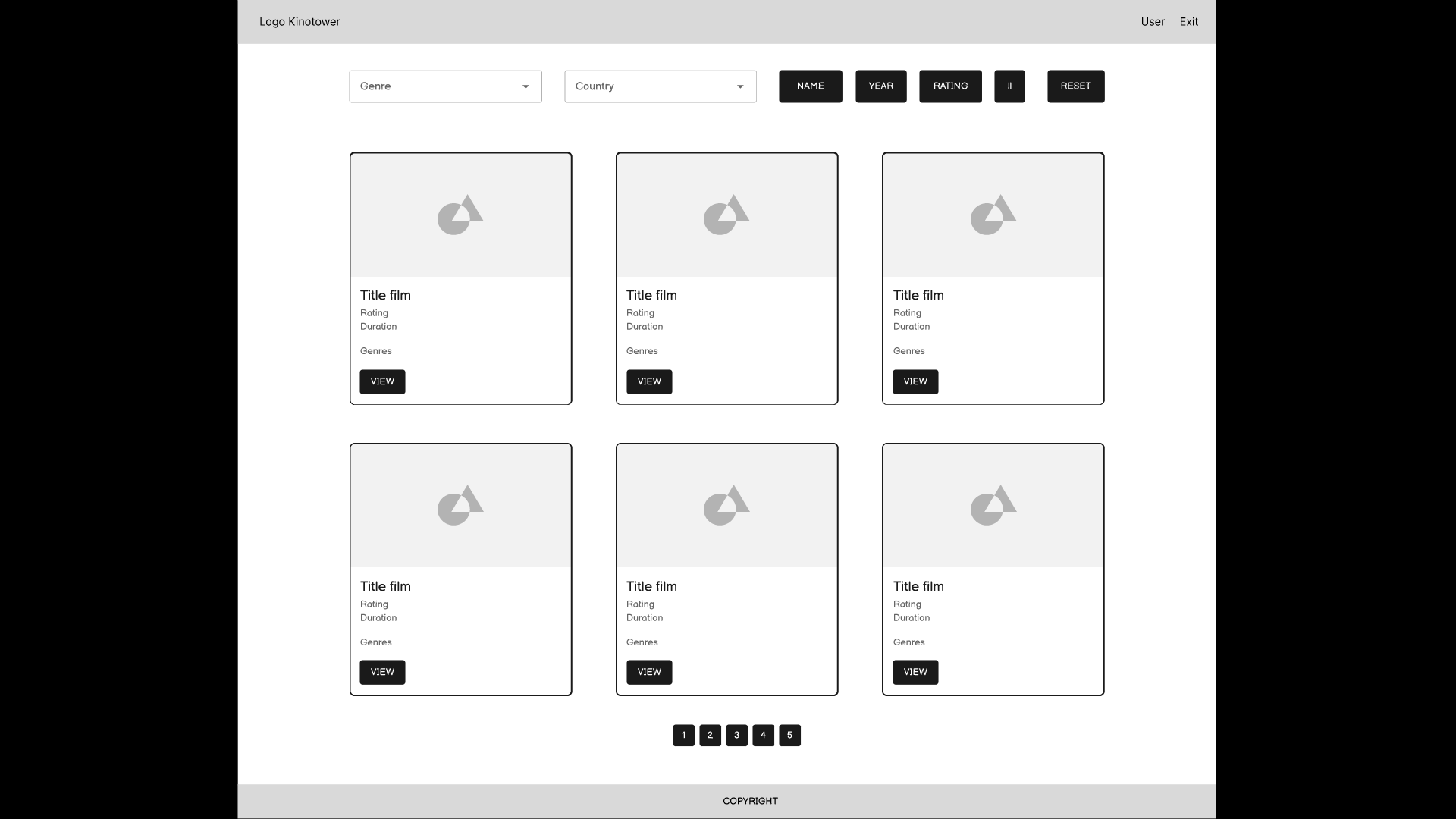
}

**2 ЭТАП СОЗДАНИЕ САЙТА**

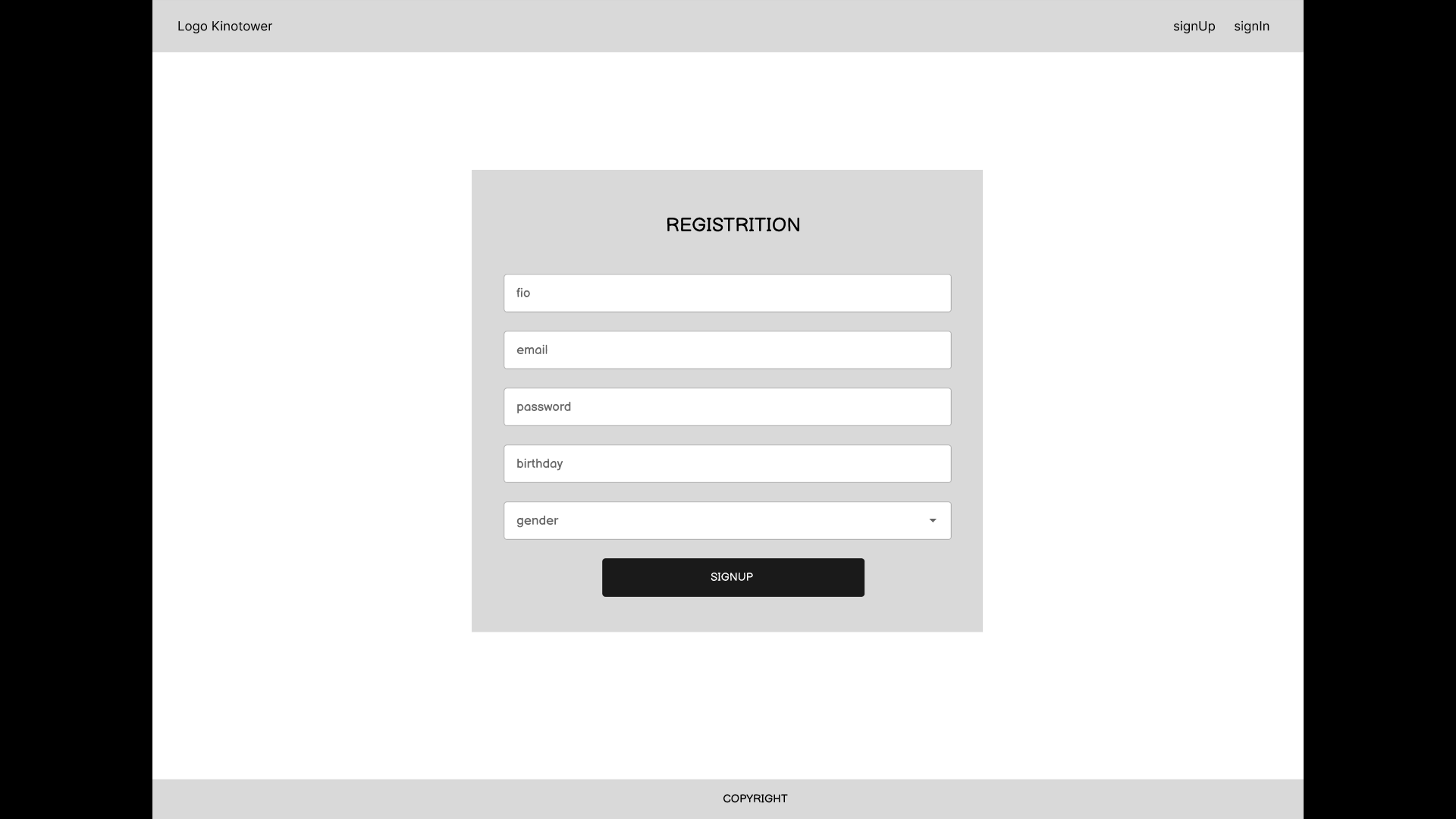
На этом этапе необходимо разработать сайт в виде SPA приложения, который использует API разработанное на первом этапе.

Для каждой страницы, которую вы должны реализовать предоставлен макет. Макеты являются черновыми руководствами. Вы можете придумать свой собственный макет, цветовую схему и пользовательский опыт. Тем не менее, вы должны придерживаться функциональности, которую демонстрируют макеты.

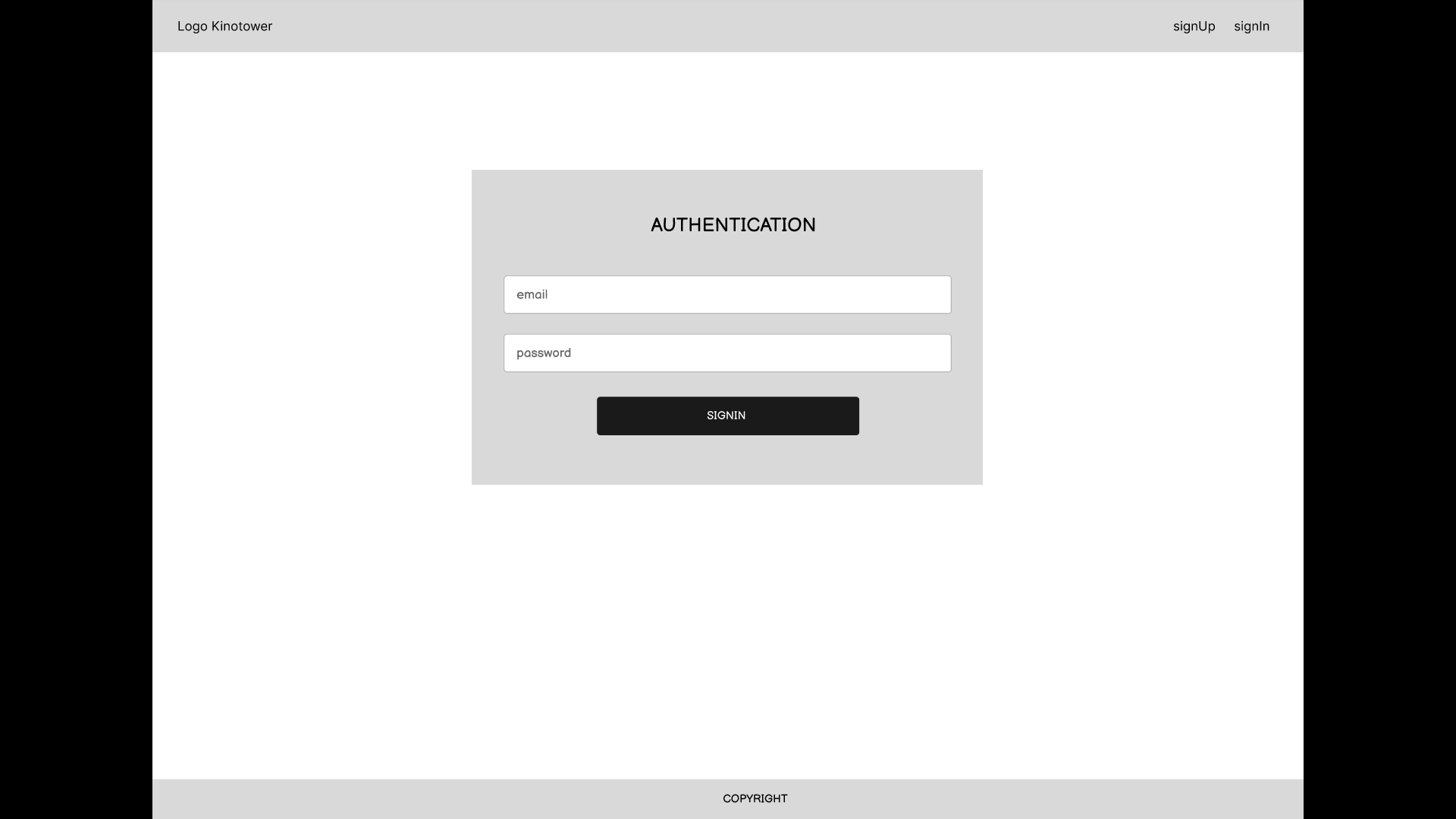
Главная страница сайта со списком фильмов



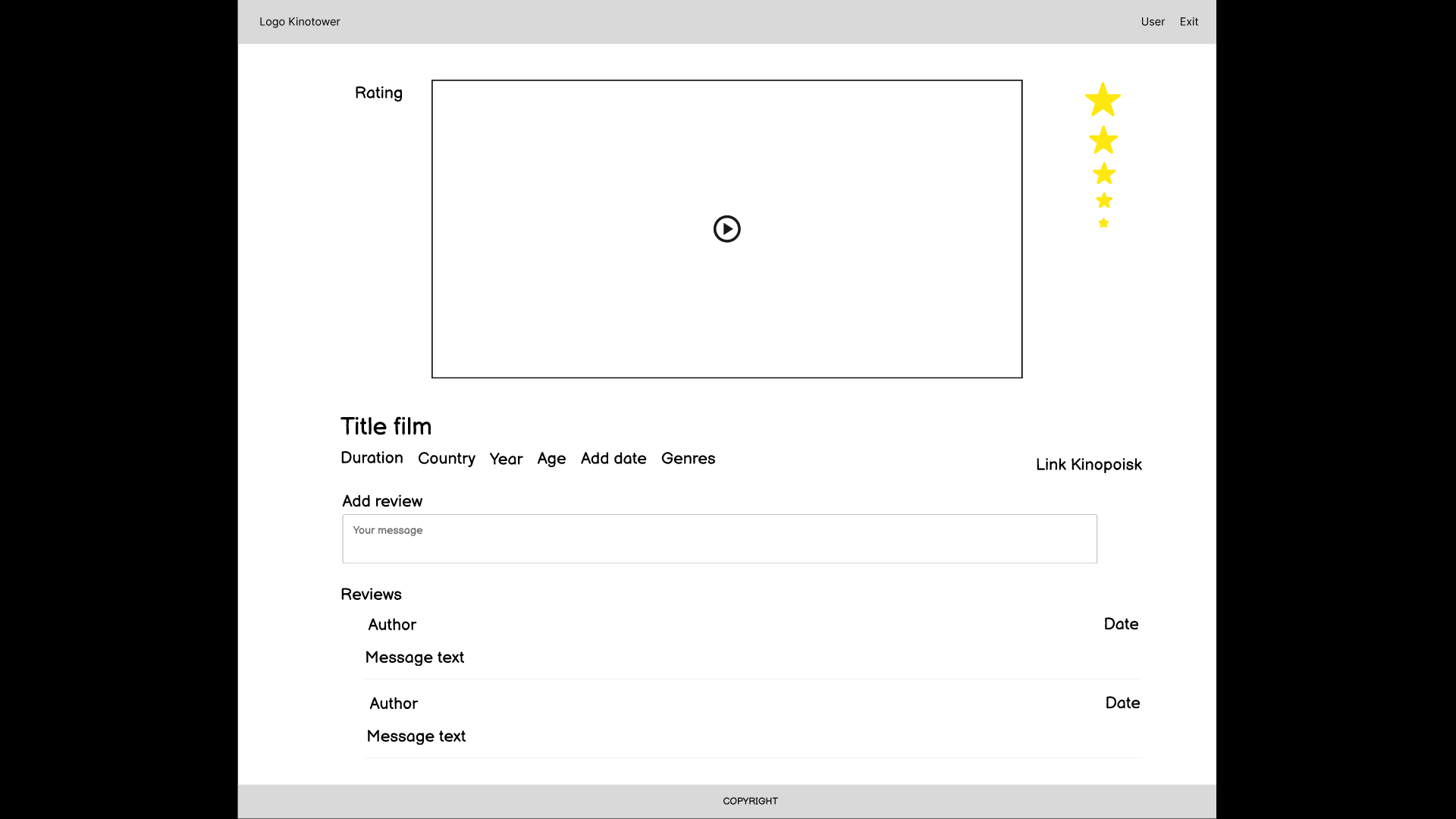
Страница регистрации пользователя



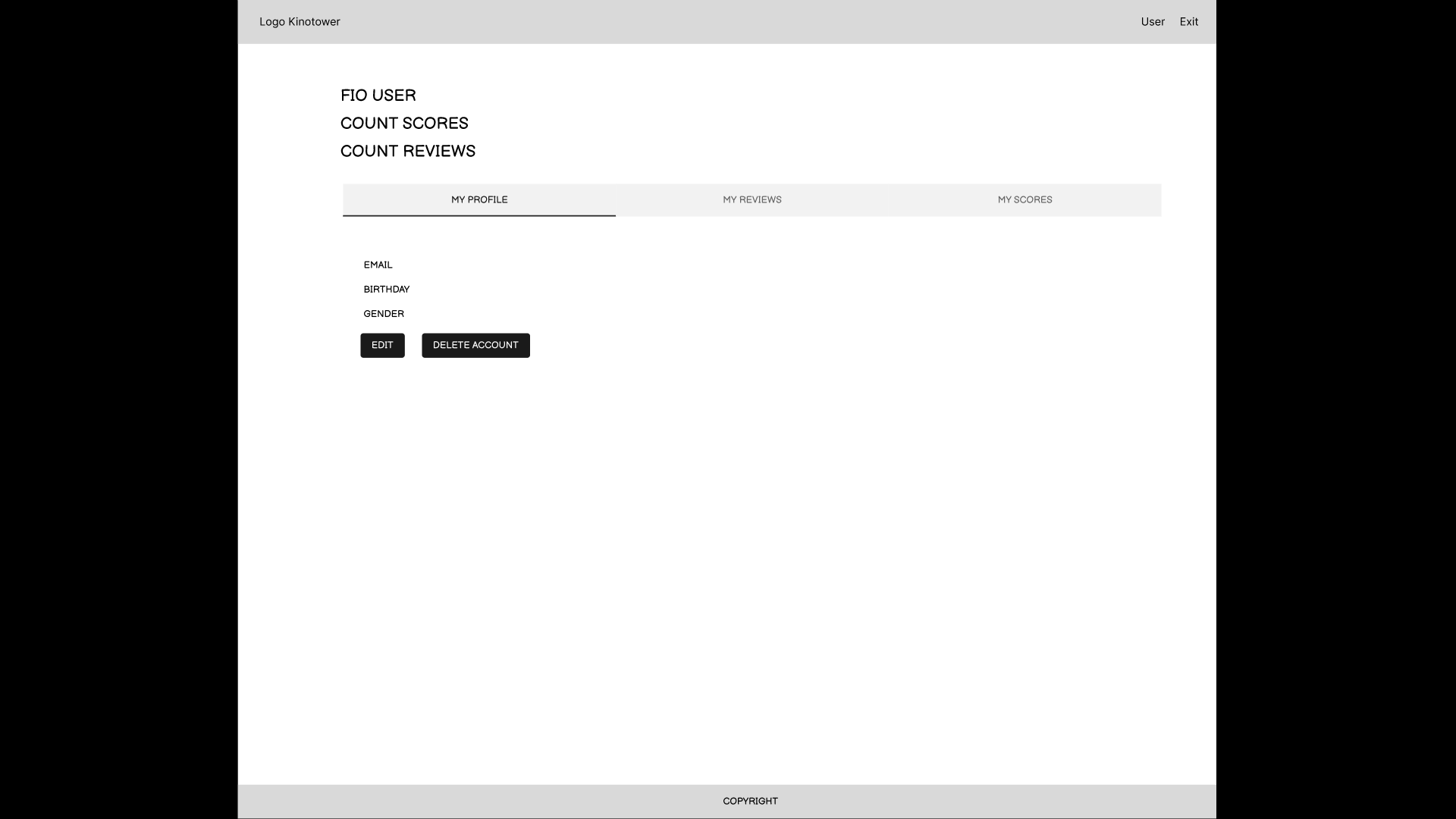
Страница входа



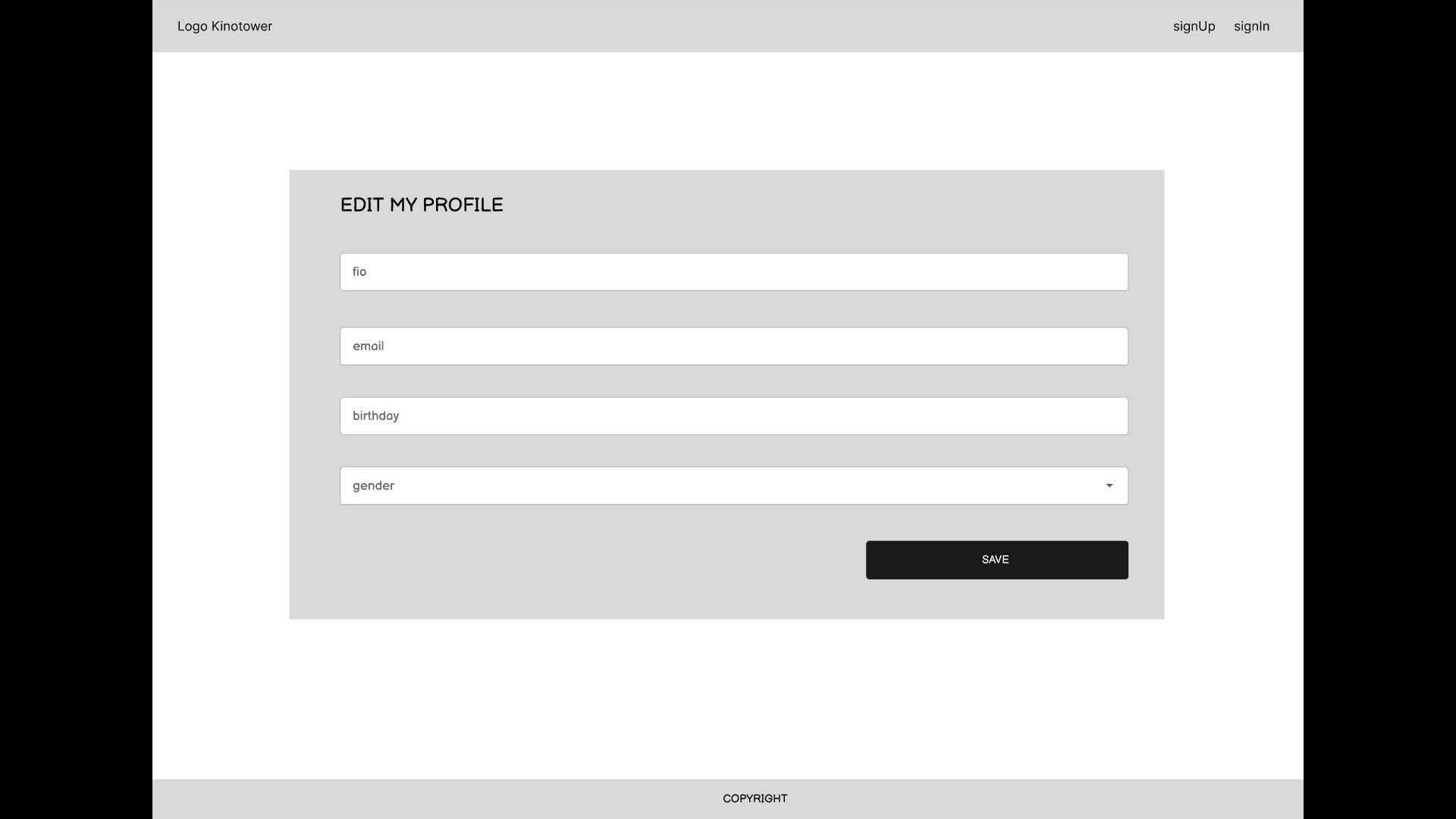
Страница просмотра фильма



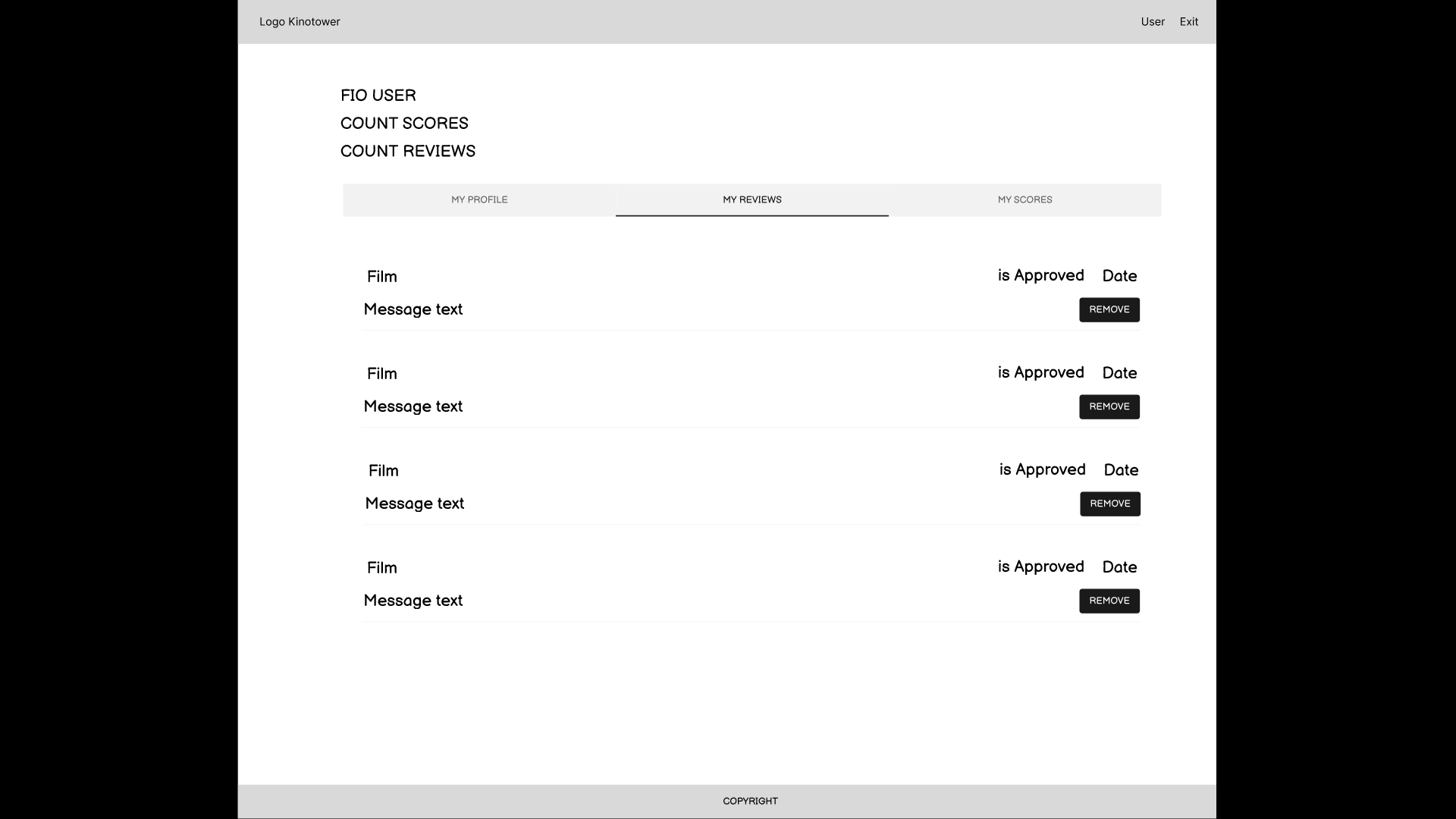
Страница данных профиля



Страница редактирования данных пользователя



Страница просмотра профиля, вкладка с отзывами пользователя



Страница просмотра профиля, вкладка с оценками пользователя

